**Commandes R\_Statistique Descriptive**

**Création et Manipulation de Données**

1. **c()** : Création de vecteurs
2. **data.frame()** : Création de dataframes
3. **rep()** : Répétition de valeurs
4. **cbind()** : Combinaison de colonnes
5. **names()** : Nommage des colonnes de dataframes
6. **as.numeric()** : Conversion en valeurs numériques
7. **read.table()** : Lecture de fichiers
8. **file.choose()** : Sélection interactive de fichier
9. **attach()** : Attacher un dataframe pour accès direct aux colonnes

**Fonctions Statistiques Descriptives**

1. **length()** : Compter le nombre d'éléments
2. **mode()** : Déterminer le type de données
3. **str()** : Afficher la structure d'un objet
4. **unique()** : Extraire les valeurs distinctes
5. **table()** : Créer un tableau de fréquence
6. **cumsum()** : Calculer les sommes cumulatives
7. **mean()** : Calculer la moyenne
8. **median(**) : Trouver la valeur médiane
9. **weighted.mean()** : Calculer une moyenne pondérée
10. **range()** : Obtenir les valeurs min et max
11. **quantile()** : Calculer les quartiles
12. **var()** : Calculer la variance
13. **sd()** : Calculer l'écart-type

**Fonctions de Manipulation Conditionnelle**

1. **ave()** : Calculer des moyennes par groupe
2. **aggregate()** : Analyses conditionnelles (moyenne, variance, somme, etc.)
3. **by()** : Alternative à aggregate pour calculs conditionnels

**Fonctions de Représentation Graphique**

1. **hist()** : Créer un histogramme
2. **boxplot()** : Créer une boîte à moustaches
3. **rug()** : Ajouter les points individuels sur un graphique
4. **par()** : Configuration des paramètres graphiques

**Fonctions Supplémentaires**

1. **prop.table()** : Calculer les fréquences relatives
2. **round()** : Arrondir des valeurs

**Opérations et Techniques Spécifiques**

1. Sélection conditionnelle : **elev[elev==4]**
2. Calcul de fréquence : **length(elev[elev==4])/length(elev)**
3. Séquence avec **seq()** : Création d'intervalles pour histogrammes

# Cours Détaillé : Commandes R pour Statistiques Descriptives

## Introduction

Ce cours couvre toutes les commandes essentielles utilisées pour la statistique descriptive en R. Chaque commande sera expliquée en détail avec des exemples concrets.

## ****1. Création et Manipulation de Données****

### **c() : Création de vecteurs**

* **Fonction** : Créer un vecteur en regroupant plusieurs valeurs.
* **Syntaxe** : vecteur <- c(valeur1, valeur2, ...)
* **Exemple** :
* tailles <- c(138, 164, 150)

print(tailles)

**Résultat** : [1] 138 164 150

### **data.frame() : Création de dataframes**

* **Fonction** : Créer une table de données (dataframe).
* **Syntaxe** : df <- data.frame(colonne1 = valeurs1, colonne2 = valeurs2)
* **Exemple** :
* eleves <- data.frame(Nom = c("Alice", "Bob"), Taille = c(150, 160))

print(eleves)

**Résultat** :

Nom Taille

1 Alice 150

2 Bob 160

### **rep() : Répétition de valeurs**

* **Fonction** : Répéter une ou plusieurs valeurs.
* **Syntaxe** : rep(valeur, fois)
* **Exemple** :

rep(5, 3)

**Résultat** : [1] 5 5 5

### **cbind() : Combinaison de colonnes**

* **Fonction** : Ajouter des colonnes à une table.
* **Syntaxe** : nouveau\_df <- cbind(df, colonne\_supplémentaire)
* **Exemple** :
* notes <- cbind(eleves, Note = c(15, 18))

print(notes)

**Résultat** :

Nom Taille Note

1 Alice 150 15

2 Bob 160 18

### **names() : Nommage des colonnes de dataframes**

* **Fonction** : Modifier les noms des colonnes.
* **Syntaxe** : names(df) <- c("colonne1", "colonne2")
* **Exemple** :
* names(eleves) <- c("Prenom", "Hauteur")

print(eleves)

**Résultat** :

Prenom Hauteur

1 Alice 150

2 Bob 160

### **as.numeric() : Conversion en valeurs numériques**

* **Fonction** : Convertir des données en valeurs numériques.
* **Syntaxe** : vecteur\_numerique <- as.numeric(valeur)
* **Exemple** :
* valeurs <- as.numeric(c("3", "5"))

print(valeurs)

**Résultat** : [1] 3 5

### **read.table() : Lecture de fichiers**

* **Fonction** : Lire un fichier texte et le convertir en dataframe.
* **Syntaxe** : df <- read.table("chemin\_du\_fichier", header = TRUE)
* **Exemple** :
* data <- read.table("data.txt", header = TRUE)

print(data)

### **file.choose() : Sélection interactive de fichier**

* **Fonction** : Ouvrir une fenêtre de sélection de fichier.
* **Syntaxe** : fichier <- file.choose()

### **attach() : Attacher un dataframe pour accès direct aux colonnes**

* **Fonction** : Permet d’accéder directement aux colonnes sans préciser le nom du dataframe.
* **Syntaxe** : attach(df)
* **Exemple** :
* attach(eleves)

print(Taille)

# ****Fonctions Statistiques Descriptives en R****

## ****1. length() : Compter le nombre d'éléments****

* **Fonction** : Retourne le nombre d'éléments dans un vecteur ou une liste.
* **Syntaxe** : length(objet)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

length(tailles)

**Résultat** : [1] 3

## ****2. mode() : Déterminer le type de données****

* **Fonction** : Indique le type de données contenues dans un objet.
* **Syntaxe** : mode(objet)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

mode(tailles)

**Résultat** : [1] "numeric"

## ****3. str() : Afficher la structure d'un objet****

* **Fonction** : Donne des informations structurées sur un objet (type, taille, contenu).
* **Syntaxe** : str(objet)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

str(tailles)

**Résultat** : num [1:3] 138 164 150

## ****4. unique() : Extraire les valeurs distinctes****

* **Fonction** : Retourne les valeurs uniques d’un vecteur.
* **Syntaxe** : unique(vecteur)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150, 138)

unique(tailles)

**Résultat** : [1] 138 164 150

## ****5. table() : Créer un tableau de fréquence****

* **Fonction** : Compte la fréquence de chaque valeur unique dans un vecteur.
* **Syntaxe** : table(vecteur)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150, 138)

table(tailles)

**Résultat** :

Copy code

138 150 164

2 1 1

## ****6. cumsum() : Calculer les sommes cumulatives****

* **Fonction** : Calcule les sommes cumulées d'un vecteur.
* **Syntaxe** : cumsum(vecteur)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

cumsum(tailles)

**Résultat** : [1] 138 302 452

## ****7. mean() : Calculer la moyenne****

* **Fonction** : Calcule la moyenne arithmétique des valeurs d’un vecteur.
* **Syntaxe** : mean(vecteur)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

mean(tailles)

**Résultat** : [1] 150.6667

## ****8. median() : Trouver la valeur médiane****

* **Fonction** : Retourne la médiane des valeurs d’un vecteur.
* **Syntaxe** : median(vecteur)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

median(tailles)

**Résultat** : [1] 150

## ****9. weighted.mean() : Calculer une moyenne pondérée****

* **Fonction** : Calcule une moyenne pondérée à partir des valeurs et de leurs poids.
* **Syntaxe** : weighted.mean(valeurs, poids)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

poids <- c(0.2, 0.5, 0.3)

weighted.mean(tailles, poids)

**Résultat** : [1] 153.2

## ****10. range() : Obtenir les valeurs min et max****

* **Fonction** : Retourne les valeurs minimale et maximale d’un vecteur.
* **Syntaxe** : range(vecteur)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

range(tailles)

**Résultat** : [1] 138 164

## ****11. quantile() : Calculer les quartiles****

* **Fonction** : Retourne les quantiles d’un vecteur (par défaut 0%, 25%, 50%, 75%, 100%).
* **Syntaxe** : quantile(vecteur, probs)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

quantile(tailles)

**Résultat** :

shell

Copy code

0% 25% 50% 75% 100%

138.0 144.0 150.0 157.0 164.0

## ****12. var() : Calculer la variance****

* **Fonction** : Retourne la variance des éléments d’un vecteur.
* **Syntaxe** : var(vecteur)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

var(tailles)

**Résultat** : [1] 169.3333

## ****13. sd() : Calculer l’écart-type****

* **Fonction** : Calcule l’écart-type des valeurs d’un vecteur.
* **Syntaxe** : sd(vecteur)
* **Exemple** :

R

Copy code

tailles <- c(138, 164, 150)

sd(tailles)

**Résultat** : [1] 13.01536

## ****3. Fonctions de Manipulation Conditionnelle****

### **ave() : Calculer des moyennes par groupe**

* **Fonction** : Calcule la moyenne d'un vecteur pour chaque groupe spécifique.
* **Syntaxe** : ave(valeurs, groupes)
* **Exemple** :
* scores <- c(1, 2, 3, 4, 3, 4, 5, 6, 7)
* genre <- c("homme", "homme", "homme", "homme", "femme", "femme", "femme", "femme", "femme")

ave(scores, genre)

**Résultat** : [1] 2.5 2.5 2.5 2.5 5.0 5.0 5.0 5.0 5.0

### **aggregate() : Analyses conditionnelles (moyenne, variance, somme, etc.)**

* **Fonction** : Effectue des calculs statistiques pour chaque groupe.
* **Syntaxe** : aggregate(valeurs ~ groupes, donnees, fonction)
* **Exemple** :
* donnees <- data.frame(Nom = c("Aira", "Aira", "Ben", "Ben"),
* Taux = c(12, 18, 53, 22))

aggregate(Taux ~ Nom, donnees, mean)

**Résultat** :

Nom Taux

1 Aira 15

2 Ben 37.5

### **by() : Alternative à aggregate pour calculs conditionnels**

* **Fonction** : Applique une fonction sur des sous-ensembles d'un vecteur divisés par groupe.
* **Syntaxe** : by(valeurs, groupes, fonction)
* **Exemple** :
* scores <- c(1, 2, 3, 4, 3, 4, 5, 6, 7)
* genre <- c("homme", "homme", "homme", "homme", "femme", "femme", "femme", "femme", "femme")

by(scores, genre, mean)

**Résultat** :

genre: femme

[1] 5

------------------------------------------------------------

genre: homme

[1] 2.5

## ****Résumé des Commandes : Fonctions de Manipulation Conditionnelle****

|  |  |  |  |
| --- | --- | --- | --- |
| **Commande** | **Fonction** | **Syntaxe** | **Exemple** |
| ave() | Moyennes par groupe | ave(valeurs, groupes) | ave(scores, genre) |
| aggregate() | Analyses conditionnelles (moyenne, somme, etc.) | aggregate(valeurs ~ groupes, ...) | aggregate(Taux ~ Nom, donnees, mean) |
| by() | Calculs conditionnels par groupe | by(valeurs, groupes, fonction) | by(scores, genre, mean) |

# ****Résumé des Commandes****

| **Commande** | **Fonction** | **Syntaxe** | **Exemple** |
| --- | --- | --- | --- |
| length() | Compte les éléments | length(vecteur) | length(c(1, 2, 3)) |
| mode() | Détermine le type de données | mode(objet) | mode(c(1, 2, 3)) |
| str() | Structure de l'objet | str(objet) | str(data.frame(x = 1:3)) |
| unique() | Valeurs distinctes | unique(vecteur) | unique(c(1, 2, 2)) |
| table() | Tableau de fréquence | table(vecteur) | table(c(1, 2, 2)) |
| cumsum() | Sommes cumulatives | cumsum(vecteur) | cumsum(c(1, 2, 3)) |
| mean() | Moyenne | mean(vecteur) | mean(c(1, 2, 3)) |
| median() | Médiane | median(vecteur) | median(c(1, 2, 3)) |
| weighted.mean() | Moyenne pondérée | weighted.mean(valeurs, poids) | weighted.mean(c(1, 2), c(0.5)) |
| range() | Min et Max | range(vecteur) | range(c(1, 2, 3)) |
| quantile() | Quartiles | quantile(vecteur) | quantile(c(1, 2, 3)) |
| var() | Variance | var(vecteur) | var(c(1, 2, 3)) |
| sd() | Écart-type | sd(vecteur) | sd(c(1, 2, 3)) |